

10/08/00
JC929 U.S. PTO

10-10-00

A₁

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

ATTORNEY DOCKET NO. 0544MH-35791

JC894 U.S. PTO
09/680603

10/08/00

In re Application of:

MARK YABLONSKI ET AL

Serial No.

Filed: 8 OCTOBER 1999

For: SYSTEM AND METHOD FOR DISPLAYING GRAPHS

TRANSMITTAL

Hon. Commissioner of Patents
and Trademarks
Washington, D.C. 20231

Sir:

Enclosed find:

1. Patent Application;
2. informal Drawings
3. Our check for \$710.00
4. Our return postcard, which we would appreciate your date-stamping and returning to us upon receipt.

CERTIFICATE OF EXPRESS MAIL 37 C.F.R. § 1.10

Express Mail No. EL 609929616 US

I hereby certify that this correspondence is being deposited with the United States Postal Service as "Express Mail Post Office to Addressee" in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on the date below.

8 Oct 2000
Date

[Signature]
Signature

The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. **50-1060**.

Please conduct all correspondence in the above matter with:

Kenneth C. Hill
HILL & HUNN, LLP.
201 Main Street, Suite 1440
Fort Worth, Texas 76102-3105
(817) 332-2113
(817) 332-2114 (fax)

Respectfully submitted,



Melvin A. Hunn, Reg. No. 32,574
Kenneth C. Hill, Reg. No. 29,650
HILL & HUNN, LLP.
201 Main Street, Suite 1440
Fort Worth, Texas 76102
817/332-2113

Attorney for Applicant

SPECIFICATION

Docket No. 0544MH-35791

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN that we, Mark Yablonski, Bradley Babin, and Weylin Debetez, residing in the State of Texas, have invented new and useful improvements in a

SYSTEM AND METHOD FOR DISPLAYING GRAPHS

of which the following is a specification:

CROSS REFERENCE TO RELATED APPLICATIONS

1 This application claims the benefit of US Provisional Application No.
2 60/158,651, filed October 8, 1999, titled AGGREGATION MECHANISM FOR
3 FUNCTIONAL GRAPHS.

BACKGROUND OF THE INVENTION

4 1. Field of the Invention:

5 The present invention relates generally to computer system displays, and
6 more specifically to a system and method for graphing functions over user
7 selectable aggregations of input data.

8 2. Description of the Prior Art:

9 The prior art has long been practiced on paper by mathematicians and
10 more recently in computer graphics user interfaces (GUIs) by software
11 engineers. The prior art graphs a function on a grid of two or three axes
12 commonly labeled X, Y, and Z. One of the axes corresponds to the function
13 output values; the other axis or axes correspond to function data inputs. The
14 axes show relatively simple measurements such as Day of Week vs.
15 Temperature. Point, line or bar displays within the grid display the function
16 values. For instance, the grid could graph average, high, or low temperature
17 functions given sets of temperatures read hourly during specific days. (See

1 Figure 1.) The prior art also includes multiple function graphs, in which the data
2 axis (such as Day of Week) is subdivided by function. For instance, the Day of
3 the Week axis could display Monday Average, Monday High, Monday Low,
4 Tuesday Average, Tuesday High, Tuesday Low, etc. The graph would display
5 three side-by-side bars for each day. (See Figure 2.)

6 The prior art also displays hierarchically organized axes measures. For
7 instance, the Day of the Week axis could display Monday morning, Monday
8 afternoon, and Monday evening. (See Figure 3.) This is no different from a ruler
9 breaking its measure into inches and further into half inches, quarter inches, etc.
10 The prior art additionally could combine multiple function graphs and
11 hierarchically organized axes to display aggregate function values. For instance,
12 the Day of the Week axis could display Monday, Monday Morning, Monday
13 afternoon, Monday evening, etc. One bar would plot Monday's aggregate
14 temperature over morning, afternoon, and evening; another would plot Monday
15 Morning's temperature, etc. Thus, the reader could see the daily aggregate
16 temperatures as well as the morning, afternoon, and evening temperatures.
17 Another way of showing aggregations is by plotting them as separate functions,
18 using different colors, shading, labeling, etc. to differentiate function values. (See
19 Figure 4.)

20 The prior art can also move the functional value axis to the grid. For
21 instance, in a two dimensional graph, the function might be on both X and Y axes
22 data, and the function value is displayed on the grid via a number or a bar scaled

1 independently from the axes scales. Often such a bar is numbered as well. For
2 example, the game Sim City™ displays a grid where the X and Y-axis
3 coordinates are longitude and latitude geographic measures. Various bars on
4 the grid show population statistics at the corresponding X and Y coordinates.
5 The heights of the bars have nothing to do with the X and Y-axes—the bars are
6 on their own scale, displaying population measures. The X and Y-axes are
7 geographic measures in terms of miles. (See Figure 5.)

8 The prior art also includes three-dimensional graphs, involving relatively
9 simple X, Y, and Z-axis measures. The graph displays a function of X and Y
10 axes data plotted against the Z-axis. (See Figure 6.) Alternatively, as in the
11 previous discussion, the function value axis is moved to the grid, displaying a
12 function of data on all three axes. (See Figure 7.) Such graphs are difficult to
13 read on paper, but software rotation techniques and 3-dimensional perspective
14 displays make this approach viable.

15 The prior art is practiced on both paper illustrations and GUIs. GUIs
16 provide various navigation capabilities such as scrolling through the axes.

SUMMARY OF THE INVENTION

1 According to the present invention, a system and method for displaying
2 graphs on a computer display organize the data and axes of the display
3 hierarchically. The user is allowed to select which levels of the hierarchy are to be
4 displayed, thereby controlling the look of the display. Data at lower hierarchical
5 levels is aggregated at the level chosen for display. By varying the chosen display
6 levels, the user changes the nature of the display on the computer monitor.

RECEIVED
"PATENT"
RECEIVED

BRIEF DESCRIPTION OF THE DRAWINGS

1 The novel features believed characteristic of the invention are set forth in the
2 appended claims. The invention itself however, as well as a preferred mode of use,
3 further objects and advantages thereof, will best be understood by reference to the
4 following detailed description of an illustrative embodiment when read in
5 conjunction with the accompanying drawings, wherein:

6 Figures 1 – 7 are graphs in accordance with prior art graphing techniques;

7 Figures 8 – 21 are graphs illustrating features of the preferred graphing
8 technique; and

9 Figures 22 and 23 are hierarchy diagrams illustrating aspects of the
10 preferred method.

DESCRIPTION OF THE PREFERRED EMBODIMENT

1 The present invention employs a computer graphics user interface (GUI)
2 to advance the state of the art in graphing functions. It improves over the prior art
3 by providing a mechanism for displaying the functional values of user-chosen
4 aggregates of the function's input data. Prior art systems can display aggregated
5 values such as the temperature on Monday, along with the unaggregated
6 values—the temperature on Monday morning, afternoon, and evening. The
7 weakness in such graphs is that the viewer cannot choose the level of
8 aggregation. In the previous example, the graph designer chose daily
9 aggregations. A viewer might instead want to see weekly aggregations of
10 various types, such as weekly average temperatures, weekly morning average
11 temperatures, etc. Alternatively, the user might want to see the graph
12 aggregated by weekdays and weekends. The present invention provides such a
13 mechanism in a graphics user interface in which the graph axes are organized in
14 hierarchies (tree structures) that the user can manipulate. One or both axes (in a
15 2 dimensional graph) can be so organized. In a three dimensional graph, one,
16 two, or all three axes can be so organized. The user can navigate the graph in
17 all manners common in the prior art (e.g. scrolling). The user can select various
18 levels of the hierarchies to see aggregate function values.

19 For example, Figure 8 shows the graph of a function of time plotting
20 values of temperature. Perhaps the function is average temperature given a set
21 of temperature readings throughout each day. The time data has been

1 organized in a time hierarchy with days at the lowest level, weeks at the next
2 level, years above the weeks level, and Time itself at the top level. The days
3 level has been selected initially, so the function values for each day are
4 displayed. The user then selects a different hierarchy level to see a different
5 level of aggregation. Figure 9 shows that the user selected the weeks level,
6 resulting in function values displayed for weekly aggregates of the input data. In
7 this example, the graph shows average weekly temperature. The user can also
8 select specific portions of a given hierarchy level to further refine the aggregation.
9 In Figure 10, the user has reselected the days level, grouped together weekdays,
10 and grouped together weekends. The user can easily study weekday vs.
11 weekend temperature variations in an application that may not have anticipated
12 such a use.

13 These time vs. temperature examples show the convenience provided by the
14 present invention. However the present invention is a key analysis tool in more
15 complex applications such as manufacturing, where the user needs to view
16 functions over complex hierarchies of data. Examples of such data hierarchies in
17 manufacturing are:

- 18 • Seller (sales force), organized hierarchically by geographic location. For
19 instance, a Seller hierarchy could consist of the entire sales force at the top
20 level. The second level could contain the worlds' continents. The third level
21 could contain countries within each continent.

1 • Product, organized by product family.. For instance, the top level of the
2 Product hierarchy could represent all products. The second level could
3 contain various product groups, and the third level could contain products
4 within each product group.

5 • Time, organized by subdivisions of time.

6 The bottom layers of such hierarchies often break into vectors of data. For
7 instance, at the bottom of the Seller hierarchy might be a vector of sales data
8 containing Planned sales, Allocated Sales, and Actual Sales. If this were plotted
9 vs. a Time hierarchy, the user could go up and down the Seller and Time
10 hierarchies to have the graph answer such complex questions such as:

11 1. What were the Allocated Sales in North America last year?

12 2. What were the Allocated Sales in the United States last year?

13 3. What were the Actual Sales in the United States the 1st quarter of last year?

14 The seller hierarchy designer has established that Planned Sales, Allocated
15 Sales, and Actual Sales are a vector of numbers to display at each cell of the
16 grid. The user however can issue instructions to compose a special function to
17 display, built on values in the vector—a feature provided in current spreadsheets.
18 For instance, the user could plot Actual Sales minus Planned Sales.

The user can additionally filter out portions of one or both hierarchies to answer such questions as:

4. What were the Allocated Sales in North America and Asia (but not Europe nor South America) last year?

5. What were the Allocated Sales in North America and Asia (but not Europe nor South America) during the 1st quarters of the last five years?

To view such numbers, the user selects nodes Europe and South America and executes a Filter command. Europe and South America then disappear from the Seller hierarchy and are placed in a small separate window, to remind the user that they've been filtered. During the rest of the session, all aggregated values normally involving Europe and South America are adjusted, deducting the numeric contributions of Europe and South America. At any time during the rest of the session, the user can click on items in that window to remove their filters.

Figure 11 plots such a sales hierarchy vs. a time hierarchy on a 3-dimensional graph. The various sales figures are displayed using the vertical axis as well as numeric labels. The following discussion will review each of the previous five questions to see how the user employs the preferred embodiment's navigation tools to get the answers. Various features of the invention will be described in this context..

1 Suppose initially the Y-axis shows Seller and the X-axis shows All Time, a 5-
2 year history time horizon. (See Figure 12.) Using a mouse or other selection
3 mechanism, the user would select Seller to break the axis into North America,
4 South America, Europe, and Asia. (See Figure 13.) The user would select Time
5 to break it into 5 previous years. (See Figure 14.) On this selection, the grid
6 changes from displaying sales figures aggregated across all of time to yearly
7 aggregated sales figures. To answer question 1, the user cross references North
8 America vs. '98 (last year) and looks at the bar displaying Allocated Sales.

9 The user then navigates to answer question 2 by selecting North America,
10 breaking it into each country, and then cross referencing the United States vs.
11 '98. (See Figure 15.) Actually, the user has two options when drilling
12 downwards in a hierarchy. In Figure 15, the user selected the option of
13 expanding the next level only under the selected North America. Figure 16
14 shows the option of expanding the entire next level (to view all countries under all
15 continents). However, by expanding only portions of a given level, the user can
16 display a comparison of sales figures for a given country vs. a given continent, or
17 the 1st quarter of this year vs. all of last year. Command modes, mouse buttons,
18 menus, or other GUI mechanisms are used to select among the two options in a
19 manner well known in the art.

20 The user then navigates to answer question 3 by selecting '98, breaking it into
21 quarters. (See Figure 17.) Since the grid is displaying various functions
22 including Actual Sales, the user does not need to do anything special to view that

number. The user simply cross-references the United States vs. '98's 1st quarter. Note that the present invention allows choice points at these selections. In this example, the GUI might ask (via popup menu, prompt, or other methods) the user whether to break '98 into quarters, weeks, months, days, etc. Instead of offering just a single hierarchy for each axis, the present invention offers multiple alternate hierarchies for each axis.

The user then navigates to answer question 4 by issuing a command to go up a level in the Seller hierarchy to display the continents. The user then selects two of those continents, North America and Asia, to get some specified mathematical combination of their function values, such as averaging or summation. (In this example, one could view the average actual sales, the summed planned sales, etc.) The grid now displays combined sales data for the two selected continents. (Such combinations can be much more efficient than the user manually summing the sales numbers for selected continents. The present invention provides spreadsheet-like capabilities so that the user can compose complex functions from atomic functions such as summation, MIN, MAX, or AVERAGE.) The user then issues a command to go up a level in the Time hierarchy (from 1st quarter of '98 to all of '98).

The user then navigates to answer question 5 by going down the Time hierarchy to the yearly quarters, then selecting the 1st quarters of the last five years to get summed sales data on those quarters.

1 In prior art graphs, the plotted function values are simply numbers. However,
2 the present invention allows for the possibility that the function values are
3 themselves hierarchically arranged numbers. The user can navigate through this
4 value hierarchy in a manner similar to the navigation through the axis data
5 hierarchies. The only difference is that the hierarchy template for such
6 navigation occurs in a legend placed separately from the graph. For instance,
7 suppose we graphed Seller hierarchy vs. Product hierarchy on a 2 dimensional
8 graph. Suppose we wanted to graph sales numbers by product by different time
9 periods (in a time hierarchy). We could change the design to employ a 3
10 dimensional graph, with Time hierarchy as the Z-axis. However, perhaps for our
11 particular data, this would be unreadable. Instead, we have the grid display the
12 sales numbers, broken out by whatever time segments of the time hierarchy the
13 user chooses. The Time hierarchy legend would provide Last 5 Years at the top
14 level, Each Year at the next level, each quarter at the next level, etc. By clicking
15 on these various levels, the grid would display one bar for all 5 years, or 5 bars
16 for each year, or 20 bars for each quarter, etc. (See Figure 18.) As with the
17 axes hierarchies, the user can perform combination and filtering on the value
18 hierarchy. For instance, he could combine the first three years' values or filter
19 out all but each year's 1st quarter values, all through mouse commands
20 performed on the value hierarchy legend.

21 Why would a user wish to create such complex functions of three data
22 hierarchies? The following are practical questions that such a graph would easily
23 answer:

1 • What is the Planned Allocation during 2nd Quarter '99 for Texas for Product
2 A?

3 • What is the Planned Allocation during 2nd and 3rd Quarters '99 for the US for
4 Product Group 100?

5 The present invention's mechanism for graphing hierarchically organized data
6 is trivially extended to accommodate other more general data organizations such
7 as lattices and directed graphs. (Here, the term "graph" refers to the data
8 structure, not the function plotting device.) A lattice is accommodated by
9 searching the lower levels of the hierarchy under a given selected level.
10 Whenever a lower level element is encountered more than once, the repeated
11 encounters are ignored. Otherwise, the data for that element would be counted
12 multiple times in aggregating the input data. A directed graph is accommodated
13 by defining at each level the set of lowest levels. As the user drills down the
14 hierarchy, the lowest levels may actually change, if the graph has a loop and the
15 defined sets of lowest levels for given selected levels differ.

16 An example of a directed graph application of this invention is a traffic
17 analysis application with a truck or train route directed graph axis and a time axis.
18 The grid might display traffic data. The route axis is organized as a directed
19 graph of cities connected by highways. One can select a given city to see
20 information about traffic going from that city to other cities along the truck route.

1 The present invention provides a spreadsheet-like functionality in which the
2 user changes a graphed value and then updates the window to display resultant
3 changes in aggregated values. For instance, suppose the Planned Allocation for
4 1st Quarter '99 for Texas for Product A is 20. This value of 20 contributes to many
5 possible aggregate values, including the Planned Allocation for 1st Quarter '99 for
6 US for Product A, which might be 150. Suppose the aggregate function is
7 summation. If the user changes the 20 value to 40 and then selects the above
8 aggregation, the grid will display 170 instead of 150. 20 is an editable value, in
9 this case, because it is a value of leaf nodes of the axis hierarchies. 150,
10 however, is not an editable value, because it is the value of non-leaf nodes of the
11 axis hierarchies.

12 The present invention combines very elegantly and powerfully with the three
13 dimensional graphing capabilities of available products such as the floor/wall/wall
14 graphing product, available from i2 Technologies. See Figures 19-21 for various
15 examples of Seller vs. Product hierarchies, integrated with the 3-D viewing
16 mechanism of that product. These illustrations show a floor-leftwall-rightwall
17 graphic architecture where the floor is the present invention, a graph of two
18 hierarchical axes data. The left wall and right wall are mathematical
19 summarizations (selected formulas such as min, max, or average) of the
20 functional (grid) values.

21 There are many potential applications outside of manufacturing. For example,
22 any organization can provide hierarchical axes of information such as an

1 organization chart vs. geographic location, and graph function values such as
2 salary data, expense data, or skills. Those interested in financial and portfolio
3 management can use industry sectors and company sizes as the hierarchical
4 axes, and plot function values of expected and actual returns, or current
5 investment allocations. Manufacturing specific data can include, for example,
6 hierarchical axes selected from parts, products, suppliers, customers, facilities by
7 geography, size or other measure, time, plans, or measurements.
8 Corresponding plotted data can include any numeric data such as inventories,
9 demand, return-on-investment, expenses, performance, allocations, and many
10 others. Those skilled in the art will appreciate that many different type of data
11 can be graphed using the techniques described herein.

12 METHOD DETAILS

13 This section describes the procedures that implement the preferred
14 embodiment of the present invention. This invention comprises a generic GUI
15 mechanism appropriate for many types of applications. An example structure for
16 using the described method is shown in Figure 24, in which the described
17 procedures are placed in a generic Graph System 100, driven by an Application
18 104 that has its own Database 106, whether memory resident or on an external
19 disk drive. However, the invention is not limited to such an organization. The
20 Graph System 100 could be implemented as non-generic software imbedded in
21 the Application 102, for example.

1 It will be recognized by those skilled in the art that the procedures
2 described herein have alternatives that could be proved to be logically identical.
3 For instance, suppose a procedure requires two steps A and B that can be
4 executed in either order without different results. The invention is not restricted
5 to the order A, then B, just because the procedure is described in the order A,
6 then B. The procedures are expressed using a number of external functions that
7 Application 102 passes to Graph System 100 for it to call at various stages of its
8 processing. Instead of such external functions, the Application 102 could pass
9 the Graph System 100 a table of results of the functions given various inputs.
10 The procedures are not expressed in an Object Oriented framework, but many
11 Objected Oriented versions of these procedures are logically equivalent to the
12 following descriptions.

13 Note that procedures expressed in terms of external functions may appear
14 to have certain performance characteristics such as quadratic performance.
15 However, use of caching by these functions can improve the performance, e.g.
16 from quadratic to linear performance.

17 **Initialize-Graph()**

18 This procedure is used by Application 102 to send Graph System 100 the
19 axis hierarchy structures, various external functions for retrieving graph values
20 and computing value aggregations, and default GUI settings.

1 Step 1. Application 102 sends Graph System 100 the number of axes to graph
2 (either 2 or 3).

3 Step 2. For each axis,

4 Application 102 traverses Database 104, informing Graph System 100 of
5 all hierarchy nodes. Each **hierarchy-node** is a record consisting of the following:

6 (1) **label**, a string describing the node's representative section of
7 the hierarchy. In a Seller hierarchy, example labels would be North
8 America, United States, Planned Sales, Actual Sales, and Seller (the top
9 of the hierarchy).

10 (2) **subhierarchies**, a list of **subhierarchy** records each consisting
11 of:

12 (a) **child-nodes**, a list of hierarchy-node records. This list is
13 never empty. When this list has one node, that node is the
14 sole subtree of this hierarchy-node. When this list has more
15 than one node, each is an alternative subtree of this
16 hierarchy-node. In such situations, the user has a choice of
17 aggregation methods. For instance, a Year could have the
18 following subhierarchies: Monthly, Weekly, and Quarterly. A
19 Salesman hierarchy could have the following subhierarchies:

1 By Location, By Product Group, By Seniority, By
2 Performance Rating.

3 (b) **selection-label**, a string used in pop-up menus or other
4 GUI mechanisms employed by Graph System to have the
5 user select among subhierarchies.

6 When this list is empty, this hierarchy-node is a leaf of the hierarchy
7 tree.

8 (3) **initial-subhierarchy**, the initially selected subhierarchy among
9 the subhierarchies. When subhierarchies of the various nodes
10 have more than one member, there are essentially multiple
11 combinations of hierarchies that Graph System could display. The
12 initial-subhierarchies of all nodes provide Graph System the initial
13 hierarchy to display.

14 (4) **Leaf?(node, selected-nodes)**, an external function that returns
15 True if 'node' is a leaf node of the hierarchy. Otherwise it returns
16 False. When the hierarchy is a tree (the most typical situation) or a
17 lattice, Leaf? should return True if its currently selected
18 subhierarchy has no child-nodes, and the selected-nodes argument
19 should be ignored. Because this case is so typical, Application 102
20 can omit this field and accept a default function from Graph System
21 100 that does this. When the hierarchy is a graph with cycles, Leaf?

needs to be computed based on selected-nodes, which is the current set of selected hierarchy nodes. Application 102 has to traverse down from selected-nodes towards 'node', deciding how far down to traverse before leafs appropriate to that set of selected nodes are reached. (The presence of cycles means there may not be actual leafs—a traversal length within such cycles has to be decided.)

Step 3. Application sends Graph System an external function **value-lookup(xnode, ynode)** if step 1 specified two axes, or **value-lookup(xnode, ynode, znode)** if step 1 specified three axes. The arguments are hierarchy-node records that are leaf nodes of the hierarchy. The function returns a value for Graph System 100 to plot at the grid's cross-reference point of xnode, ynode, and (if present) znode. For instance, in Figure 11, suppose our xnode is the "planned" node under North America and our ynode is '95. Value-lookup applied to these two nodes would return 3.

Note that if the hierarchy is a cyclical graph, value-lookup has to return a value for every combination of nodes among those returned by the various BOTTOMS functions.

If Application 102 wishes editing capabilities on the values returned by value-lookup, Application Sends Graph System 100 an external function **value-edit(xnode, ynode)** or **value-edit(xnode, ynode, znode)**. This function handles

1 the I/O needed to retrieve the new value from the user and then changes the
2 corresponding value that is returned by value-lookup on these same arguments.

3 Step 4. Application sends Graph System a set of **Aggregation-Method** records,
4 each consisting of:

5 (1) **aggregate-value(values-list)**, an external function. Values-list
6 is a list of values over which this function computes and returns an
7 aggregate value. The list might contain numbers, and the
8 appropriate aggregation computation might be SUM or AVERAGE.
9 The value returned by aggregate-values and the members of the
10 values-list must have the same data type as that returned by the
11 value-lookup function of step 3. Graph System 100 will use this
12 function to plot aggregated values whenever non-leaf members of
13 the hierarchy are selected. (For leaf members, Graph System 100
14 can simply use value-lookup.) Aggregate-values should return
15 first(values-list) whenever values-list has only one member.

16 (2) **method-label**, a string used in pop-up menus and other GUI
17 mechanisms for the user to select their desired aggregation
18 method.

19 In many applications, only one Aggregation-Method record will be
20 provided, leaving no choice for the user. The most common aggregate-
21 value function is SUM. Another common function is AVERAGE.

1 The remaining sections will refer to the user's currently selected
2 aggregation method by its contained function, aggregate-value(values-
3 list).

4 Step 5. Application 102 informs Graph System 100 about how to present and
5 scale the graph, using methods known in the art. For instance, plotted values
6 may be presented a bars, lines, points, etc. Presentation includes coloring,
7 shading, labeling, axes scales, scrolling control, and other GUI issues.

8 Application 102 informs Graph System 100 of the nature (data type) of
9 each cell value and exactly how cell values are to be displayed. In the
10 simplest case, cell values are numbers. However, as described earlier,
11 cell values might be vectors or even hierarchies (trees) of numbers. They
12 might be symbolic values. Graph System 100 provides special formatting
13 options for each type. (The procedures required to handle cell vectors
14 and cell hierarchies are described below.)

15 Step 6. Graph System 100 determines the nature of the hierarchy using known
16 graph traversal methods in the computer science sub-fields of data structures
17 and graph theory. Here "graph" is used in its mathematical sense. The nature of
18 the hierarchy is commonly a tree structure but could be a lattice or directed
19 graph. If not a tree, Graph system 100 knows in procedures such as **Plot-**
20 **Values** to look for repeated nodes during aggregation.

Step 7. Application 102 informs Graph System 100 of the initially selected nodes for each axis. Commonly each axis hierarchy is a tree and the application sets the top or bottom nodes of each hierarchy.

Bottoms(node)

This function is used internally by Graph System 100 to determine the leaf nodes under 'node'. 'node' may be several hierarchy levels above the leafs, it might be the parent of leafs, or it might itself be a leaf. A set of the leafs is returned. The procedure is:

Let S = an empty set that prohibits adding duplicate members into itself.

If node.leaf?(node, selected-node)

Then

Add node to S.

Else

For each node CHILD in node's currently selected subhierarchy's child-nodes list,

If CHILD is not in **filtered-nodes**

Add members of Bottoms(child) into S

Return S.

Selected-nodes is the global list of selected axis hierarchy nodes.

Filtered-nodes is a procedure discussed below. Bottoms can be optimized by passing S as a second argument to Bottoms, which then populates S instead of creating and returning its own set.

Plot-Values()

At all times, the user has selected one or more nodes in the X axis hierarchy, one or more nodes in the Y axis hierarchy, and (if there is a Z-axis), one or more nodes in the Z axis hierarchy. Whenever the user changes the selected nodes, Plot-Values is called to determine the values to display in the grid area of the graph. For every combination of X-axis, Y-axis, and Z-axis nodes, Graph System has to plot a value on the grid. If a combination contains only leaf nodes, Graph System 100 can simply call value-lookup(xnode, ynode, znode) for the plotted value. However, when a combination contains at least one non-leaf node, Graph System 100 must plot an aggregation of the combinations of underlying leaf node values. For instance, suppose we have the following leaf-node combinations and values:

value-lookup(xnode1, ynode1) = 10

value-lookup(xnode1, ynode2) = 20

value-lookup(xnode2, ynode1) = 40

value-lookup(xnode2, ynode2) = 30

If xnode1, xnode2, ynode1, and ynode2 are all selected, then the above four values are all displayed on the grid. Now suppose the following: SUM is the user's currently selected aggregate-value(value-list) function, and xnode1 and xnode2 are under parent node Xnodes in the X axis hierarchy. If the user selects Xnodes, consequently deselecting xnode1 and xnode2, Graph System removes the four plotted values and plots the following two values:

1 1. the value of Xnodes vs. ynode1, which is the aggregate of the values of
2 the leaf nodes under Xnodes (xnode1 and xnode2, obtained via
3 BOTTOMS(Xnodes)) vs. ynode1. Specifically, the computation is:

4 aggregate-value({value-lookup(xnode1, ynode1), value-lookup(xnode2,
5 ynode1))})

6 = aggregate-value({10, 40})

7 = 50

8 2. the value of Xnodes vs. ynode2, which is the aggregate of the values of
9 the leaf nodes under Xnodes (xnode1 and xnode2, obtained via
10 BOTTOMS(Xnodes)) vs. ynode2. Specifically, the computation is:

11 aggregate-value({value-lookup(xnode1, ynode2), value-lookup(xnode2,
12 ynode2))})

13 = aggregate-value({20, 30})

14 = 50

15 We've stepped through the plot-values algorithm on a particular example.

16 Now the formal algorithm is as follows:

17 Clear the grid of the currently displayed values.

18 If any top node of any axis hierarchy is in filtered-nodes, return.

19 Let xnode be each selected node in the X axis hierarchy.

20 Let ynode be each selected node in the Y axis hierarchy.

1 Let znode be each selected node in the Z axis hierarchy.

2 The following is executed on each combination of xnode, ynode, and
3 znode:

4 Let value-components be an empty list whose members are the
5 same type as value-lookup.

6 Let xleaf be each node in the set returned by BOTTOMS(xnode).

7 Let yleaf be each node in the set returned by BOTTOMS(ynode).

8 Let zleaf be each node in the set returned by BOTTOMS(znode).

9 The following is executed on each combination of xleaf, yleaf, and
10 zleaf:

11 PUSH value-lookup(xleaf, yleaf, zleaf) ONTO value-
12 components.

13 Plot at the grid location corresponding to the cross-reference point
14 of xnode, ynode, and znode the value of aggregate-value(value-
15 components).

16 If there is no Z-axis, Graph System 100 simply executes an alternate
17 version of the above algorithm in which all references to Z-axis, Znode, and Zleaf
18 are removed.

19 This procedure can be optimized by avoiding calls to aggregate-value
20 whenever a combination of xnode, ynode, and znode are leaf nodes of their
21 respective hierarchies. If all three are leaf nodes, the algorithm just plots value-
22 lookup(xnode, ynode, znode). If no Z axis exists, this principle still applies and
23 the algorithm just plots value-lookup(xnode, ynode).

Drill-Down(node)

'node' is a currently selected hierarchy node. This procedure is called when the user chooses to select the immediate children of 'node'. The procedure is:

1. Remove 'node' from the set of selected nodes.
2. Add node's children nodes onto the set of selected nodes.
3. Refresh plotted values by calling plot-values().

Drill-Down(list-of-nodes)

'list-of-nodes' is a list of currently selected hierarchy nodes. This procedure is called when the user chooses to select the immediate children of each node in list-of-nodes. This is useful for efficiently drilling down from an entire hierarchy level. The algorithm for this procedure is:

1. Remove each node in list-of-nodes from the set of selected nodes.
2. Add each children node of each node in list-of-nodes onto the set of selected nodes.
3. Refresh plotted values by calling plot-values().

Drill-Up(node)

'node' is the ancestor of some currently selected hierarchy node. This procedure is called when the user chooses to select 'node', consequently deselecting any of its descendent nodes. The algorithm is:

- 1 1. Traverse all nodes lower in node's hierarchy using known methods in
- 2 computer science, removing them from the set of selected nodes.
- 3 2. Add node onto the set of selected nodes.
- 4 3. Refresh plotted values by calling plot-values().

5 **Combine(nodes-list, newlabel)**

6 This procedure is used to allow the user to modify an existing axis
7 hierarchy to view value aggregations other than those displayable through the
8 existing hierarchy. 'nodes-list' is a list of nodes in the same hierarchy, all having
9 a common ancestor among the currently selected node subhierarchies. Let us
10 call the common ancestor Anode. Anode is typically the direct parent of all
11 members of nodes-list. This algorithm temporarily deletes all nodes in the
12 hierarchy between Anode and nodes-list, making the nodes in nodes-list children
13 of Anode. Anode may have other children nodes not in nodes-list; these are left
14 untouched.

15 For example, see the previous discussion (Figure 10) where the user
16 created weekday and weekend aggregate nodes. There, Week was Anode and
17 Days were the nodes in nodes-list. No nodes existed between Weeks and Days
18 in the hierarchy. The user (via GUI commands in the Graph System) combined
19 Monday, Tuesday, Wednesday, Thursday, and Friday nodes as the children of a
20 new node whose label is "Weekdays". Weekdays became a child of the Week
21 node. The user then combined Saturday and Sunday nodes as the children of

1 the new node whose label is "Weekend". Weekend became another child of the
2 Week node.

3 In place of the hierarchy with Week at the top and seven days at the
4 bottom, the user received a three level hierarchy with Week at the top, Weekdays
5 and Weekend underneath, Monday through Friday under Weekdays, and
6 Saturday and Sunday under Weekend. Suppose (as shown in Figure 10) the
7 graph showed many weeks, rather than one week. The user (through convenient
8 GUI commands) could execute these combine operations once over all weeks in
9 the graph display.

10 The basic Combine algorithm is the following:

11 1. For all nodes between Anode and nodes-list, remove their links in the
12 hierarchy, storing them in case the user wishes the hierarchy to be restored.
13 If any were selected, unselect them and select the nodes in nodes-list.

14 2. Create a new node New and add it to Anode's children nodes of its currently
15 selected subhierarchy. Set its label to newlabel. (In the weekday/weekend
16 example, newlabel would be "weekdays" or "weekends".) Set its children
17 nodes to nodes-list. Since there is only one subhierarchy, its selection-label
18 field is irrelevant.

19 Set New's Leaf?(node, selected-nodes) function to a function which
20 ignores its arguments and returns False.

1 3. Set New's aggregation-function to Anode's aggregation-function.

2 4. Refresh plotted values by calling plot-values().

3 **Uncombine(nodes-list)**

4 This procedure is used to undo the Combine operation previously done on
5 nodes-list. The method is:

6 1. If the parent of nodes-list is a selected-node, unselect it and select the nodes
7 in nodes-list.

8 2. Destroy the parent of nodes-list, which is referred to as 'New' in Combine step
9 4.

10 3. Restore the links removed in Combine step 1.

11 4. Refresh plotted values by calling plot-values().

12 **Filter(node, replot?)**

13 This procedure is called when the user wishes to filter 'node' out of the graph,
14 as if it were not even in the Application. The algorithm is:

15 1. add node to a filtered-nodes set. Filtered-nodes is a global variable.

16 2. Let PARENT be node's parent node.

17 If PARENT's currently selected subhierarchy's child-nodes are all in filtered-
18 nodes,

19 Filter(PARENT, False)

1 3. If replot? Is True, call plot-values().

2 Step 2 recognizes that if all of a nodes children are filtered, the node itself is
3 filtered. We go ahead and filter it to cut off other algorithms' tree traversals early.
4 This is most important in plot-values, to keep a filtered subtree from being printed
5 along its axis.

6 Unfilter(node, replot?)

7 This procedure is called when the user wishes to no longer filter out 'node'.

8 The algorithm is:

9 1. If node is not in the filtered-nodes set, return.

10 2. remove node from filtered-nodes.

11 3. For each node CHILD in node's currently selected subhierarchy's child-
12 nodes,

13 Unfilter(CHILD, False)

14 4. If replot? Is True, call plot-value().

15 Step 2 makes it easy for a user to unfilter an entire subtree of nodes.

16 Cell Vectors and Cell Hierarchies

17 In the procedures described thus far, it will have been noted that various
18 data types must match. Value-lookup's return value, aggregate-value's return
19 value, and aggregate-value's arguments must all match type. Cells on the

graph's grid have values of this type. In Initialize-Graph's step 5, Application 102 tells Graph System 100 how to format this value type. For instance, if the type is numeric, perhaps a bar with or without an accompanying printed number is the format. If the value is a vector of numbers, perhaps a row of bars positioned against each other is the format (as shown in Figure 11).

Vectors fit into the current algorithms seamlessly. Aggregate-value just takes a list of vectors as argument and returns a vector containing aggregated numbers. For instance, if the argument list contained the two vectors $\langle 1, 3, 4 \rangle$ and $\langle 2, 4, 6 \rangle$, the aggregate-value function should return the vector $\langle 3, 7, 10 \rangle$. In Figure 11, thus, Planned Sales, Allocated Sales, and Actual Sales are not hierarchy nodes at all. Instead, they label the cell vector values. North America, South America, Europe, and Asia are the leaf hierarchy nodes. The formatting of their labels along the Seller Hierarchy axis would include the printing of "Planned Sales", "Allocated Sales", and "Actual Sales."

The discussion relating to Figure 18 presented the usefulness of the cell data type being a hierarchy (such as a tree of numbers). Such cell hierarchies allow the user to legibly plot functions of three hierarchies such as Seller, Product, and Time. With a cell hierarchy, the values returned by value-lookup and aggregate-values and the values plotted on each cell of the grid are all trees of numbers or other data. An example aggregation of two cell hierarchy values is shown in Figure 22. In this example, the aggregation function is the summation of numbers in the trees. Why bother with a tree representation, when we could

1 have used cell vectors to store $\langle 1, 3, 2, 5 \rangle$ and $\langle 0, 1, 0, 3 \rangle$ and defined their
2 aggregate to be the vector $\langle 1, 4, 2, 8 \rangle$? Because the cell hierarchy permits the
3 user to traverse (via drilling up, drilling down, and filtering) and manipulate (via
4 combine and uncombine) the cell hierarchy. (These algorithms on cell
5 hierarchies are almost identical to those of the axis hierarchies.) Drilling up one
6 level in the Figure 22 cell hierarchy would change the cell displaying the values
7 1, 4, 2, and 8 to displaying 5, 10. (5 is the aggregation of 1 and 4; 10 is the
8 aggregation of 2 and 8.) Drilling up another level leaves the cell displaying 15
9 (the aggregation of 1, 4, 2, and 8).

10 Initialize-graph's Step 5 is the step where Application 102 informs Graph
11 System 100 about the cell hierarchy. It passes the structure of the hierarchy as a
12 tree of hierarchy-node structures. This tree provides the tree structure and the
13 labels of leaf and nonleaf nodes of the tree, which Graph System 100 needs to
14 display its cell hierarchy in a legend. The user will click on this legend to issue
15 commands such as drilling up and down the cell hierarchy, selecting alternate
16 subhierarchies, combining, and filtering.

17 To handle cell hierarchy commands, Graph System 100 reuses slightly
18 altered versions of most of the previously presented algorithms. For Drill-Down,
19 Drill-Up, Combine, and Uncombine on cell hierarchies, the algorithms are all
20 identical to those for axis hierarchies, except that the cell hierarchy versions
21 modify a global set of cell hierarchy nodes, rather than a set of axis hierarchy
22 nodes. The latter has been referred to as "selected nodes". For the remainder of

1 this discussion, the former will be referred to as **cell-hierarchy-selected-nodes**.

2 For example, the version of Drill-Down(node) for cell hierarchies is:

3 1. Remove 'node' from cell-hierarchy-selected-nodes, the set of selected cell
4 hierarchy nodes.

5 2. Add node's children nodes onto cell-hierarchy-selected-nodes.

6 3. Refresh plotted values by calling plot-values().

7 The cell-hierarchy-selected-nodes specify how to format out all the trees of
8 values at each point on the graph's grid. For instance, if the cell value is the 1-4-
9 2-8 tree in Figure 22 and the top node of the cell hierarchy is the only member of
10 cell-hierarchy-selected-nodes, the format function should compute the aggregate
11 of 1-4-2-8 (using aggregate-value) and print that value (15). If instead the two
12 middle hierarchy nodes were selected, then it should print the values 5, 10.

13 The value formatting logic sent by Application 102 to Graph System 100 in
14 initialize-graph step 5 must take the tree of values returned by aggregate-value
15 and examine the cell hierarchy and cell-hierarchy-selected-nodes to see how to
16 format out the tree on the graph's grid. The procedure is as follows:

17 **Plot-cell-value(hnode, vnode)**

18 {hnode is the top node in the cell hierarchy. Vnode is the top node of the
19 tree of values we are plotting.}

20 IF hnode is in cell-hierarchy-selected-nodes

1 THEN
2 Let L be an empty list.
3 Traverse leafs of vnode, collecting their values in L.
4 Plot the value returned by aggregate-value(L).
5 ELSE
6 FOR EACH child HC of hnode paired with the corresponding child VC of
7 vnode,
8 Plot-cell-value(HC, VC).

9 Vnode itself might be a leaf, in which case the traversal contains just vnode.

10 Since the algorithm of this procedure may be executed over many cells,
11 speed optimizations are useful. The cell hierarchy could be traversed at the
12 start, collecting the traversal paths and corresponding aggregate-value calls to
13 make for all of the cell values to be formatted. Given the cell hierarchy in Figure
14 23, we would traverse A, B, and C nodes only once. Then, for each cell value, we
15 would know to get the leaf values in a first-order traversal and then call
16 aggregate-value({first-leaf, second-leaf}) and aggregate-value({third-leaf, fourth-
17 leaf}). If cell value trees are stored via vectors, walking the leafs in first-order is
18 extremely efficient.

19 **Editing**

20 Application 102 may be designed to operate similarly to a spreadsheet,
21 with the user changing various values and seeing resultant changes to various

1 computations in the GUI. In the case of the present invention, the user would
2 have a way of editing values retrieved by value-lookup, and on each edit, Graph
3 System would update the graph. For instance, if the graph were currently
4 displaying a summation style aggregate of four values 10,20,40,30, which is 100,
5 and the user changed the first value from 10 to 20, the graph would change to
6 display 110.

7 The simplest, non-integrated method to implement such editing in the
8 present invention is for the user to change the underlying values feeding value-
9 lookup via the Application's own editing tools, and then for the user or Application
10 102 itself to have Graph System 100 call plot-values.

11 The more convenient, integrated method is to click on a value plotted
12 between axis hierarchy leaf nodes in Graph System 100 and type in a new value.
13 Such functionality is enabled by Application 102 in Step 3 of the initialize-graph
14 algorithm.

15 As described previously, the preferred method integrates well with the
16 three dimensional graphing capabilities of products such as the floor/wall/wall
17 graphing product available from i2 Technologies. This product uses a
18 Floor/Wall/Wall (FWW) display system, to which the Graph System 100 has
19 connections. The Floor pane of FWW holds Graph System's grid and hierarchy
20 axes display. (Alternatively, one of the two walls could hold these things.
21 Typically, the primary graph data is displayed on the floor, and the walls are used

1 to show summaries.) Whenever Graph System 100 calls plot-values(), it sends
2 FWW the new Floor display and any updates to make on its walls.

3 The described system is usable with any data that can be expressed
4 hierarchically. By ordering the data in this manner, and using the described
5 procedures, a user can gain significant useful control over the display of data. As
6 is well known, well displayed data can assist the user in understanding its
7 implications, and the system and method of the present invention provides a
8 significant step forward in the art.

9 While the invention has been particularly shown and described with
10 reference to a preferred embodiment, it will be understood by those skilled in the art
11 that various changes in form and detail may be made therein without departing from
12 the spirit and scope of the invention.

1 What is claimed is:

2 1. 1. A system for displaying graphical information, comprising:

3 a display driver for making graphical information visible to a user;

4 data organized along at least one axis in a hierarchical manner; and

5 a graphing system connected to the data and to the display driver for

6 displaying at least one axis of data, in a graph having at least two dimensions, as

7 aggregated data in accordance with a selected level within the hierarchy.

8

1 2. A method for displaying data, comprising the steps of:
2 displaying a graph having at least two dimensions;
3 organizing at least one axis of the display in a hierarchical manner, so that
4 data displayed along such at least one axis is selectably aggregated to be
5 displayed.

1 3. The method of Claim 2, further comprising the step of:
2 allowing a user to select a level of the axis to be displayed, wherein the
3 selectable aggregation level displayed is that chosen by the user.

1 4. The method of Claim 3, wherein, in response to the user changing the
2 aggregation level, a different aggregation level is displayed.

1 5. The method of Claim 2, wherein the graph includes data organized in a
2 hierarchy along two axes of the display.

1 6. The method of Claim 5, wherein the display is a 3-dimensional graphical
2 display of data.

1 7. The method of Claim 1, wherein, in response to a user selection, a portion
2 of a dimension displaying aggregated data is aggregated at a different level from
3 the remaining portions of that dimension.

1 8. The method of Claim 1, wherein selected elements of a hierarchical
2 dimension are hidden from the display.

1 9. The method of Claim 1, wherein the displayed graph is organized in three
2 dimensions, with two dimensions forming a floor of a display, and both being
3 organized as hierarchies.

1 10. The method of Claim 9, wherein data is displayed in the third dimension.

2 11. The method of Claim 10, wherein the data is organized as a hierarchy,
3 and wherein the data is displayed as a selected hierarchy level.

4 12. The method of Claim 11, further comprising the steps of:

5 receiving from the user input selecting a hierarchy level for displaying the
6 data; and

7 displaying the data at the user selected level.

ABSTRACT

1 A system and method for displaying graphs on a computer display organize
2 the data and axes of the display hierarchically. The user is allowed to select which
3 levels of the hierarchy are to be displayed, thereby controlling the look of the
4 display. Data at lower hierarchical levels is aggregated at the level chosen for
5 display. By varying the chosen display levels, the user changes the nature of the
6 display on the computer monitor.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215

Fig 1

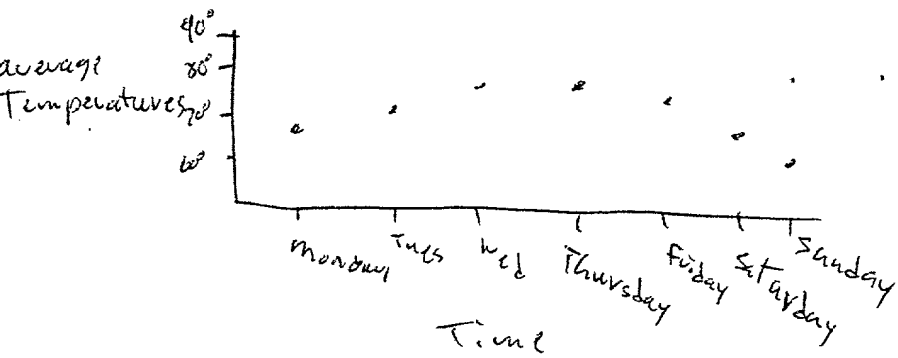


Fig 2

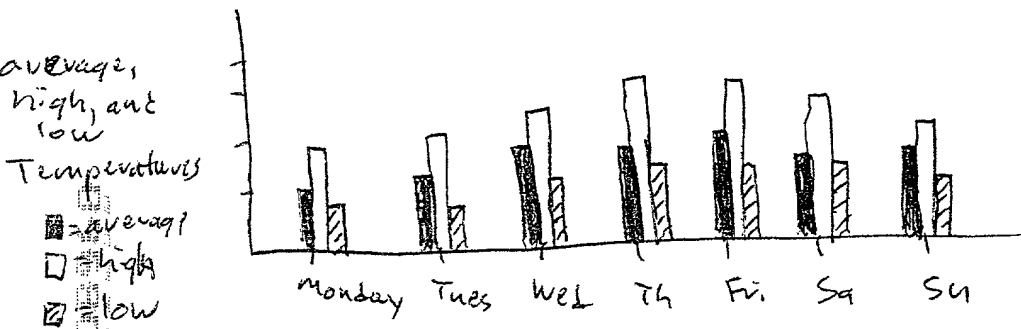


Fig 3

average
temperatures

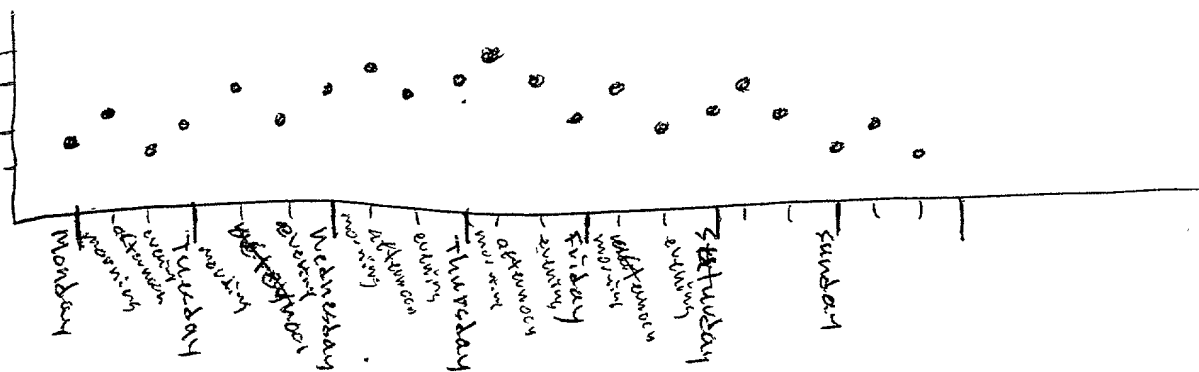


Fig 4

average
temperatures

— morning,
afternoon,
evening
— = daily
average

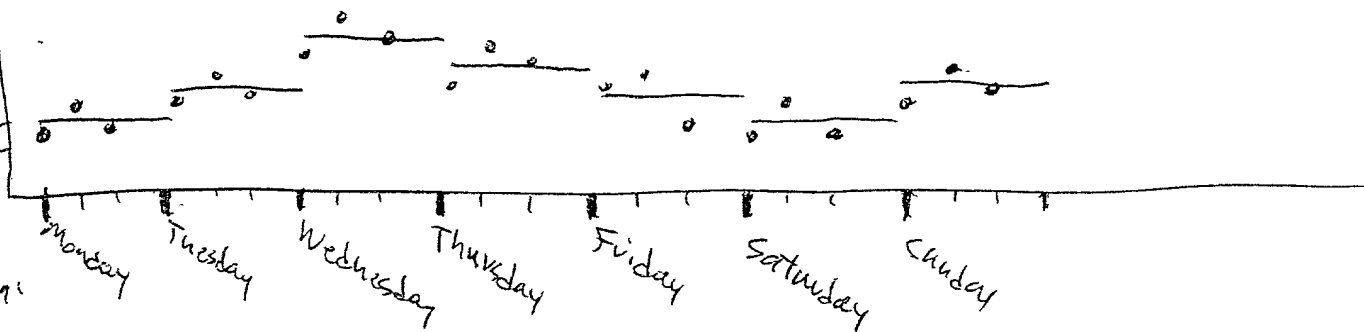
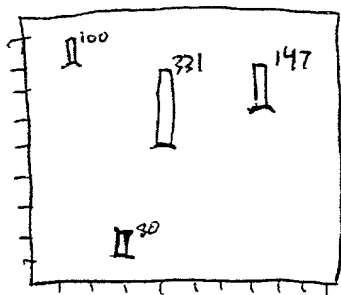


Figure 5
mockup of
Sim City (tm)
Latitude
(y axis)



Longitude (X axis)

Populations
in thousands
at various
coordinates

Figure 6

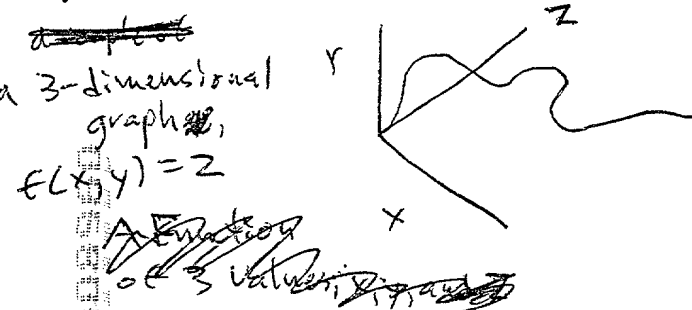


Figure 7 a 3-Dimensional graph,
 $f(x,y,z) =$ plotted
numeric
values

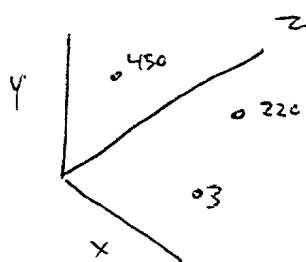


Figure 8

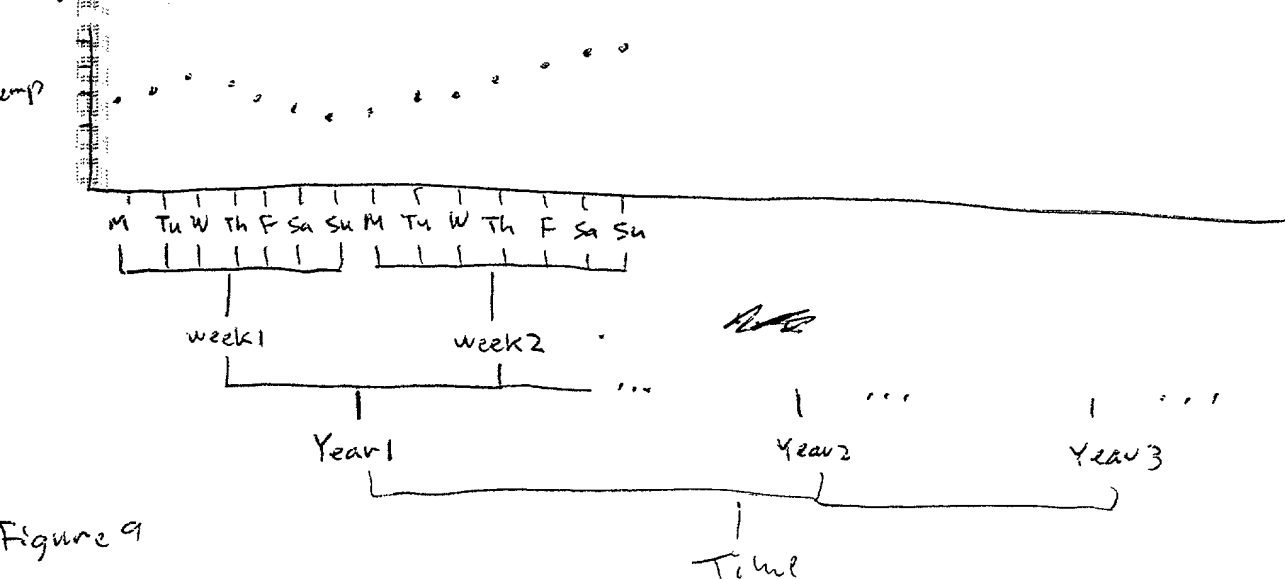


Figure 9

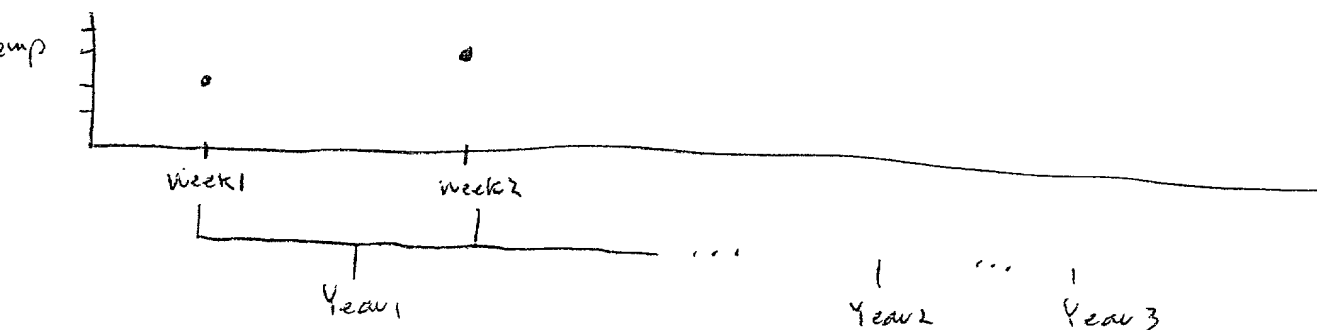


Figure 10

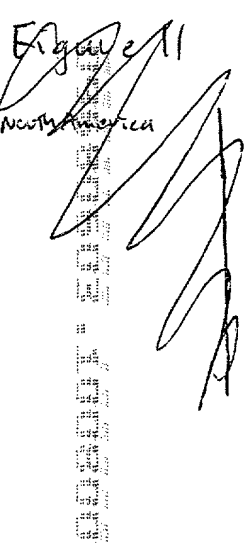
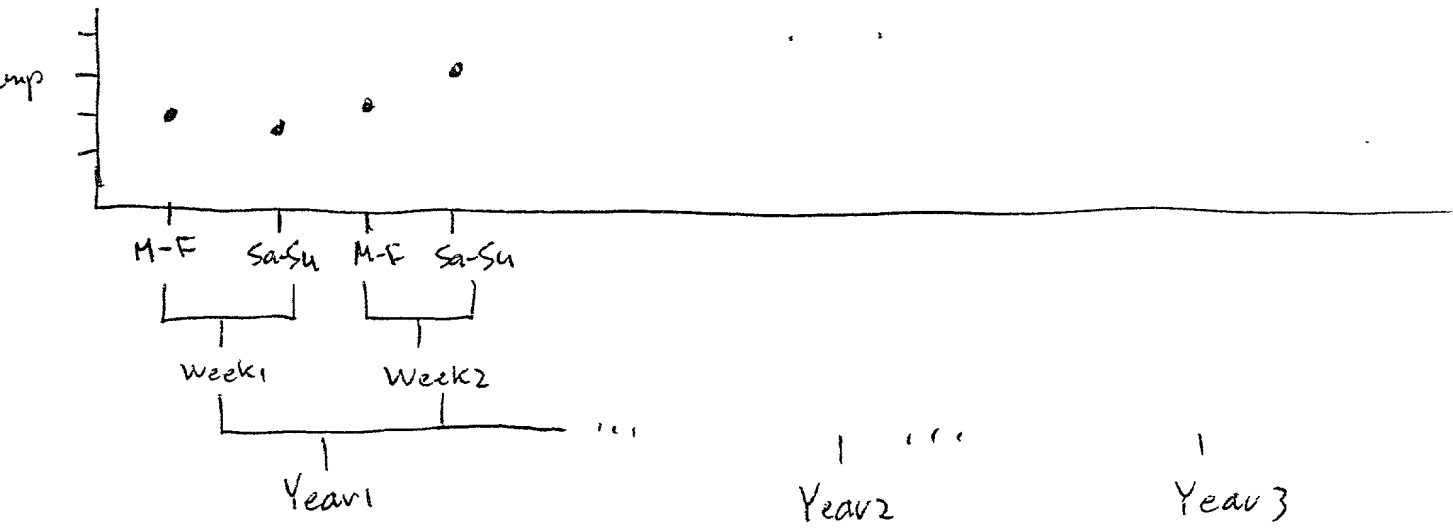


Figure 11

Sales Planned,
Allocated,
and
Actual
over time

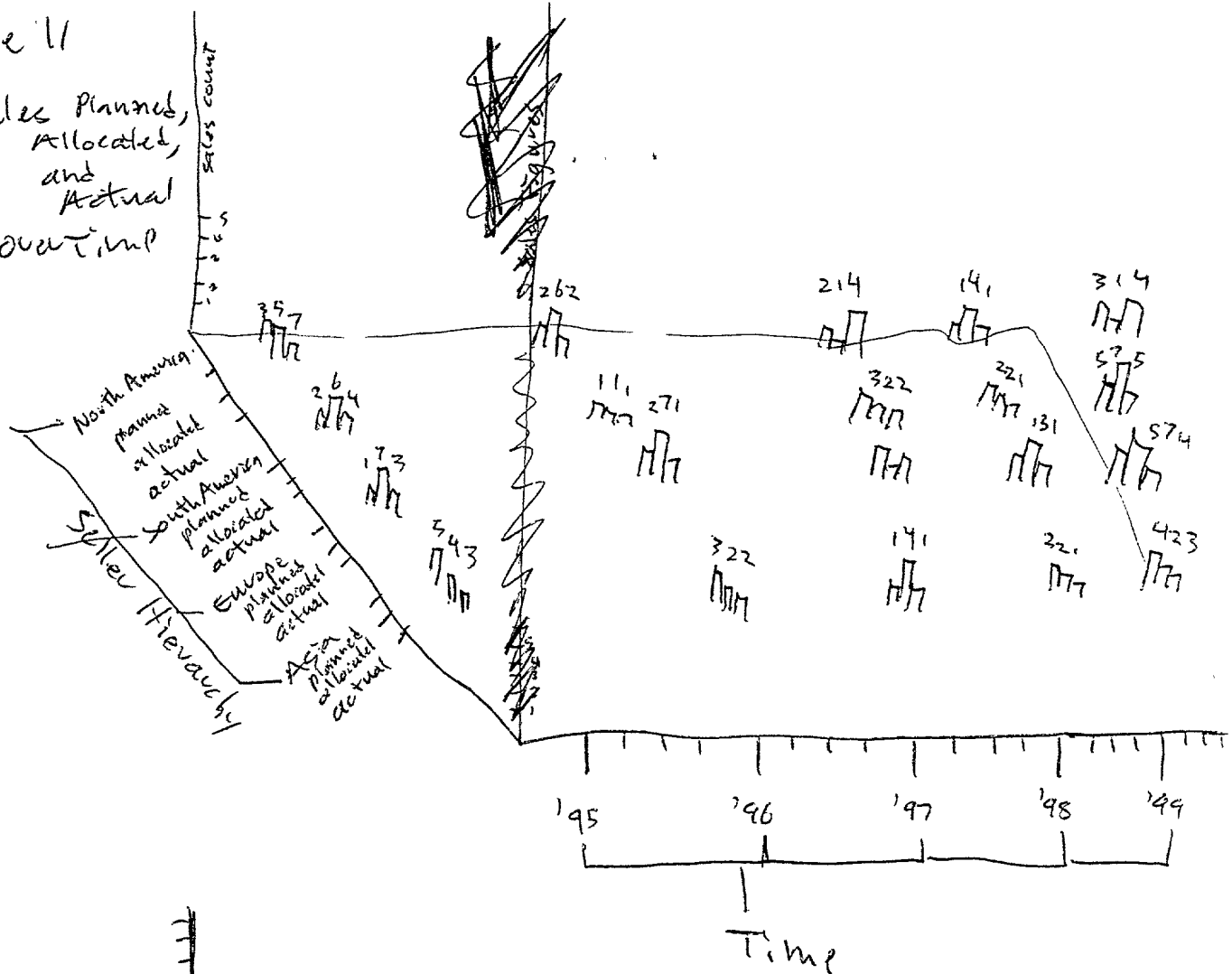


Figure 12

(4 axes)

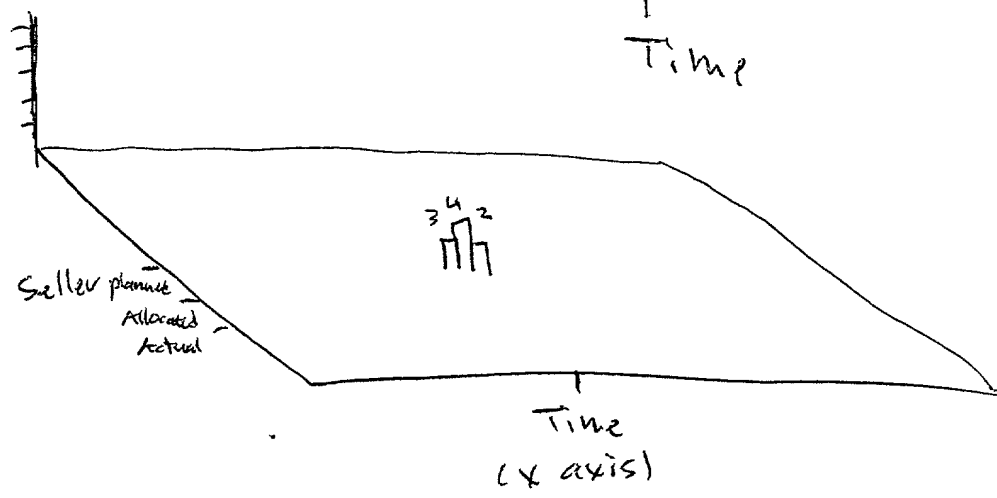


Figure 13

seller

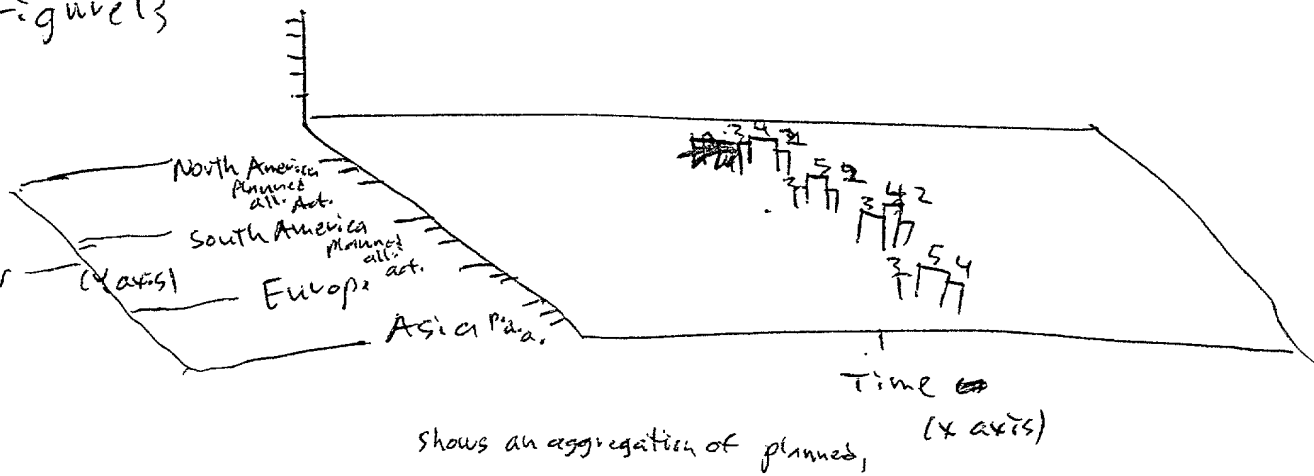


Figure 14

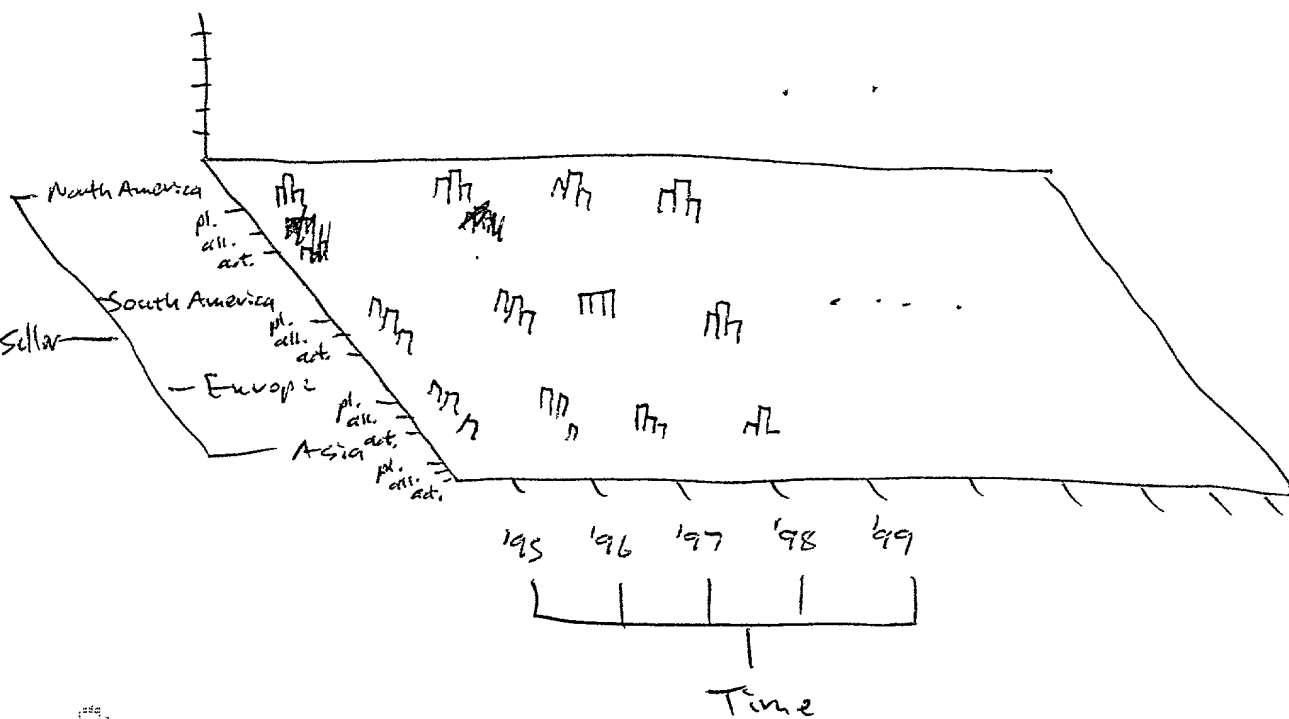


Fig. 15

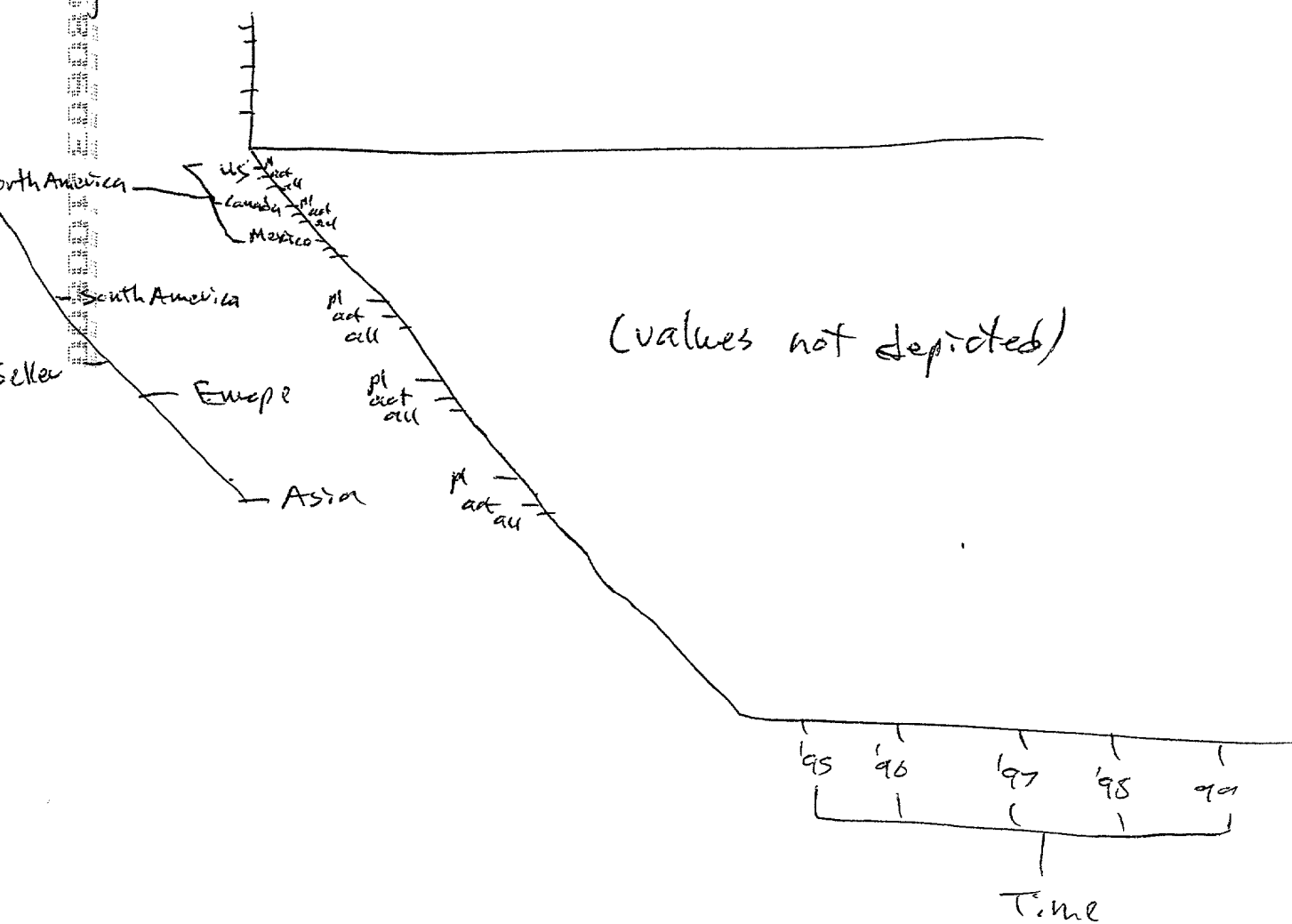


Figure 16

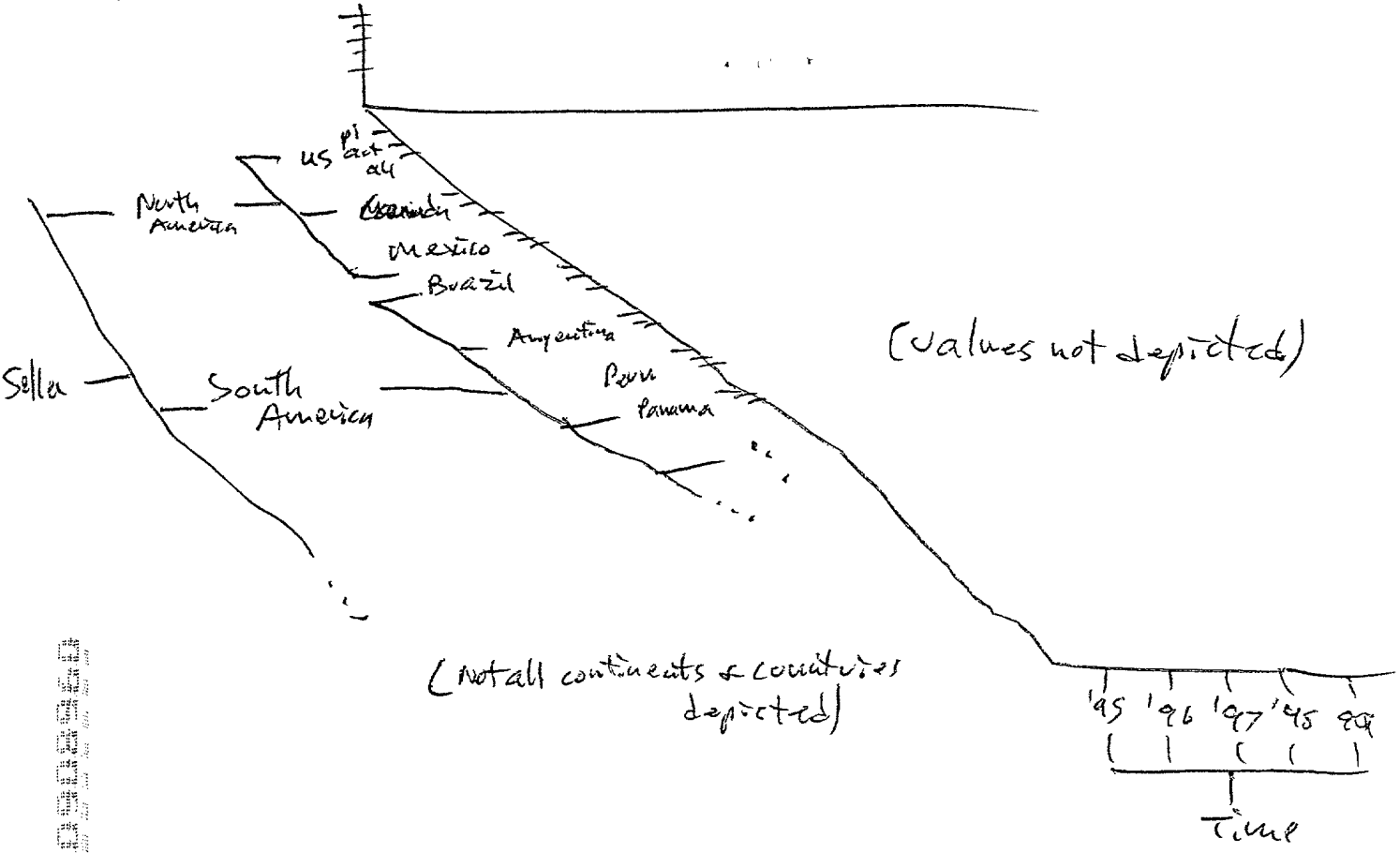
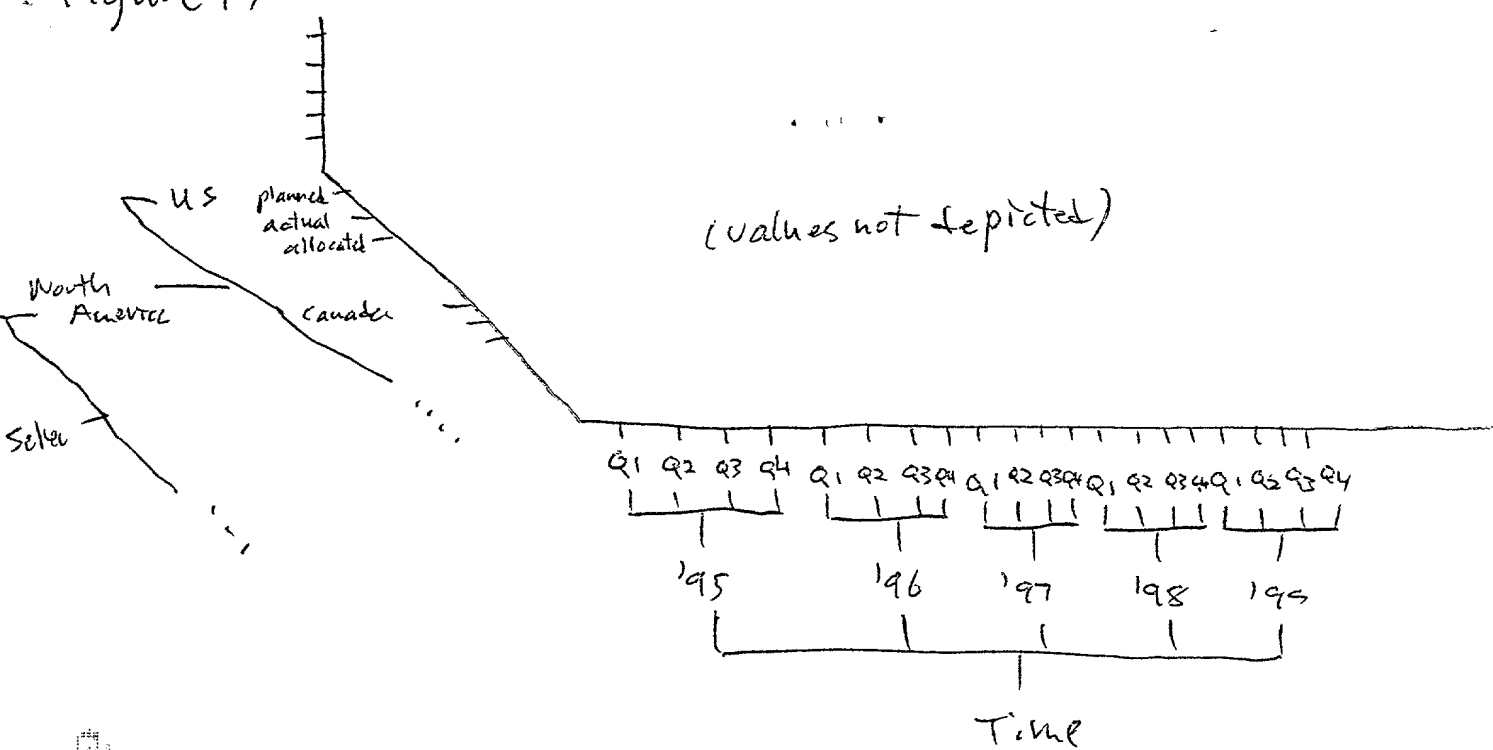


Figure 17



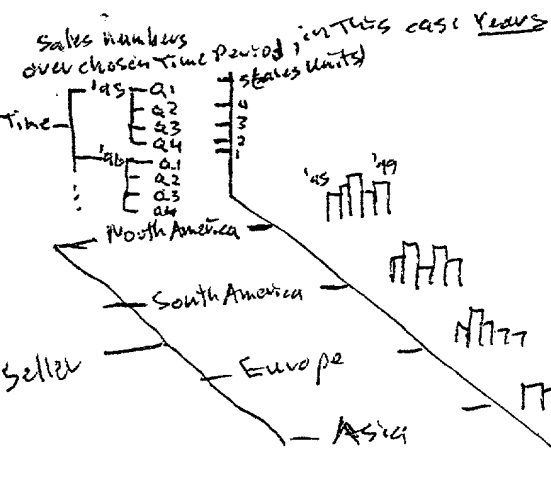
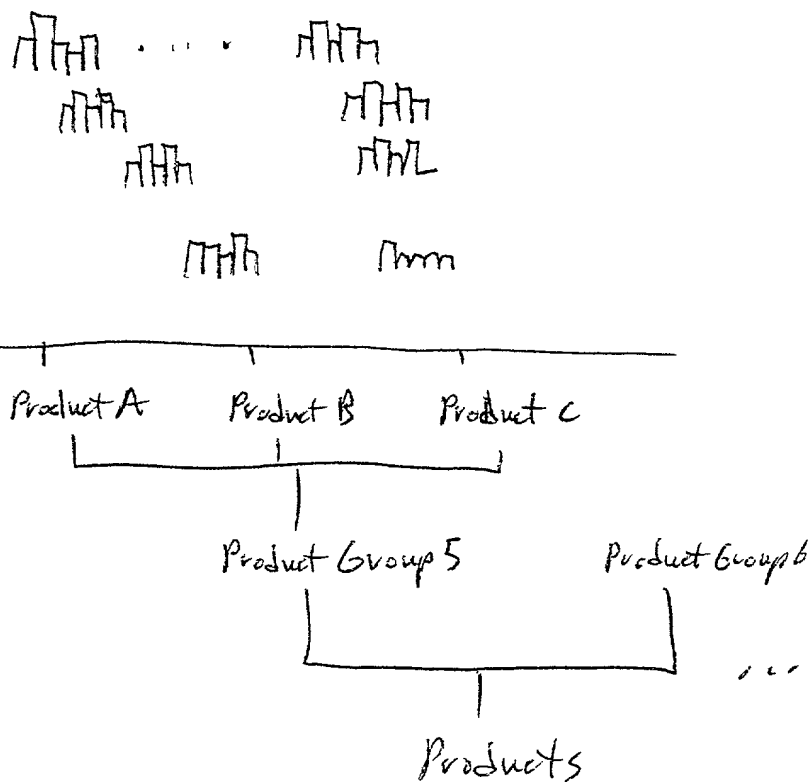


Figure 18



Figures 19-21 are the screen shots.

Figure 19

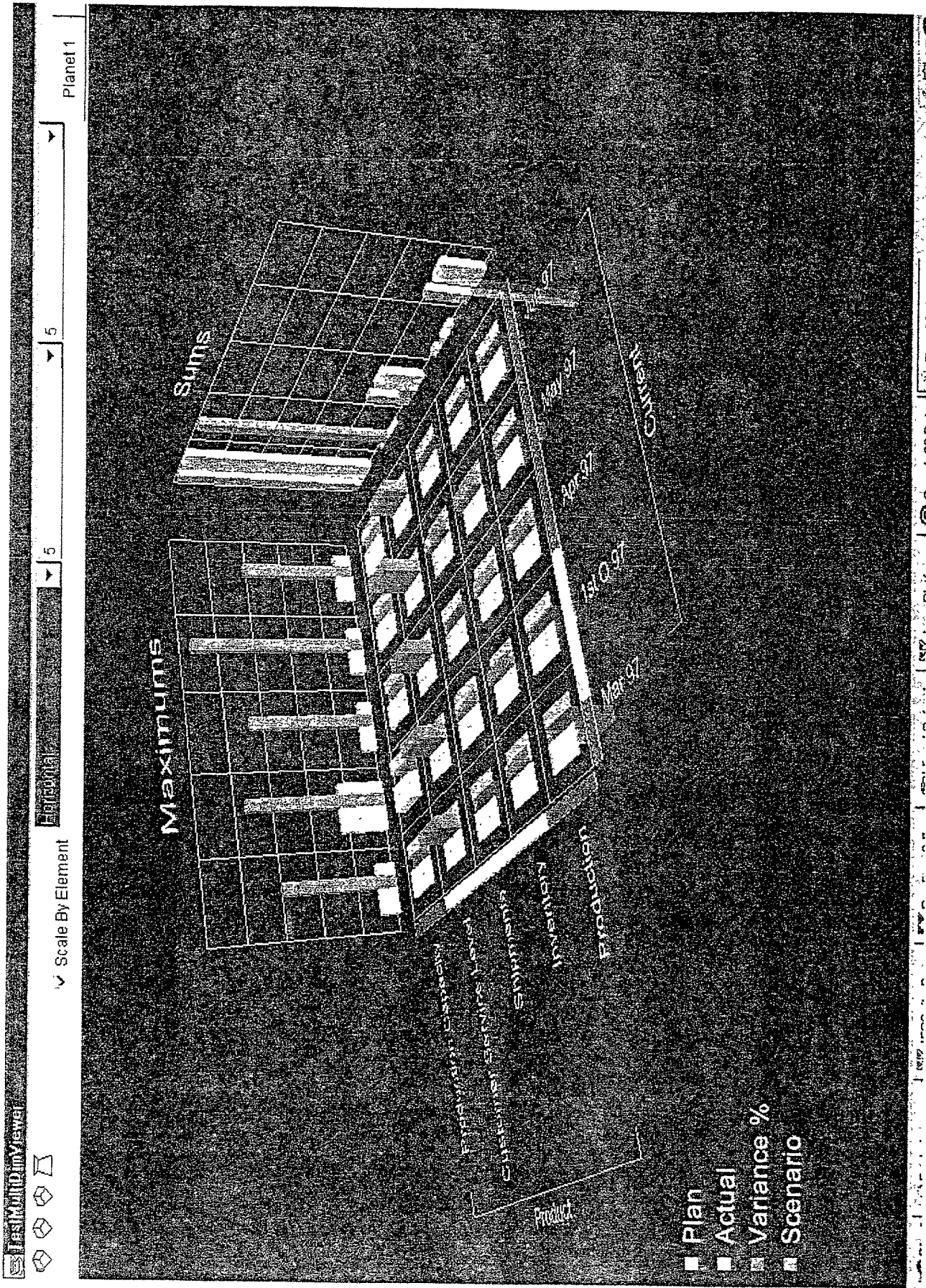
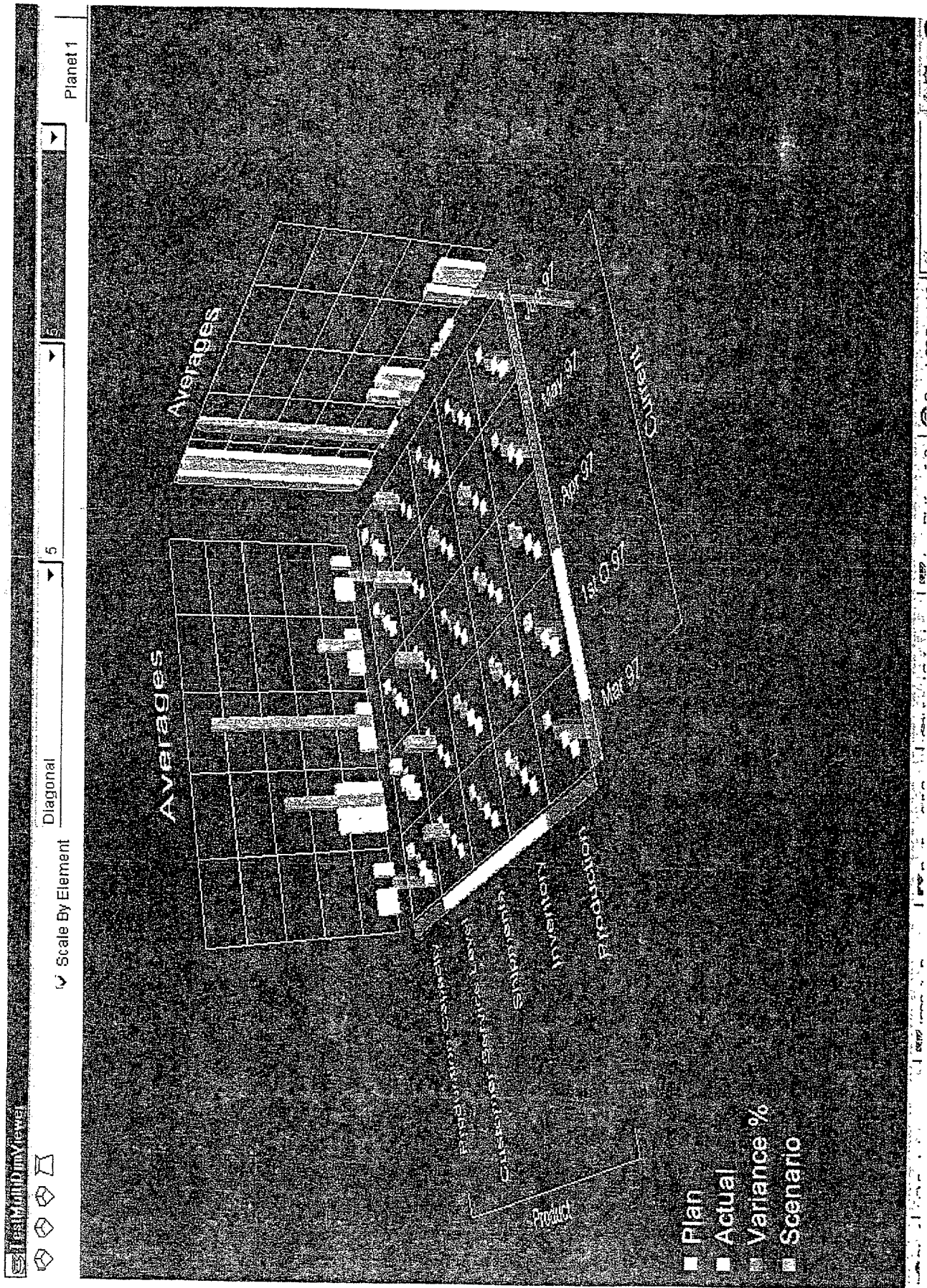


Figure 20



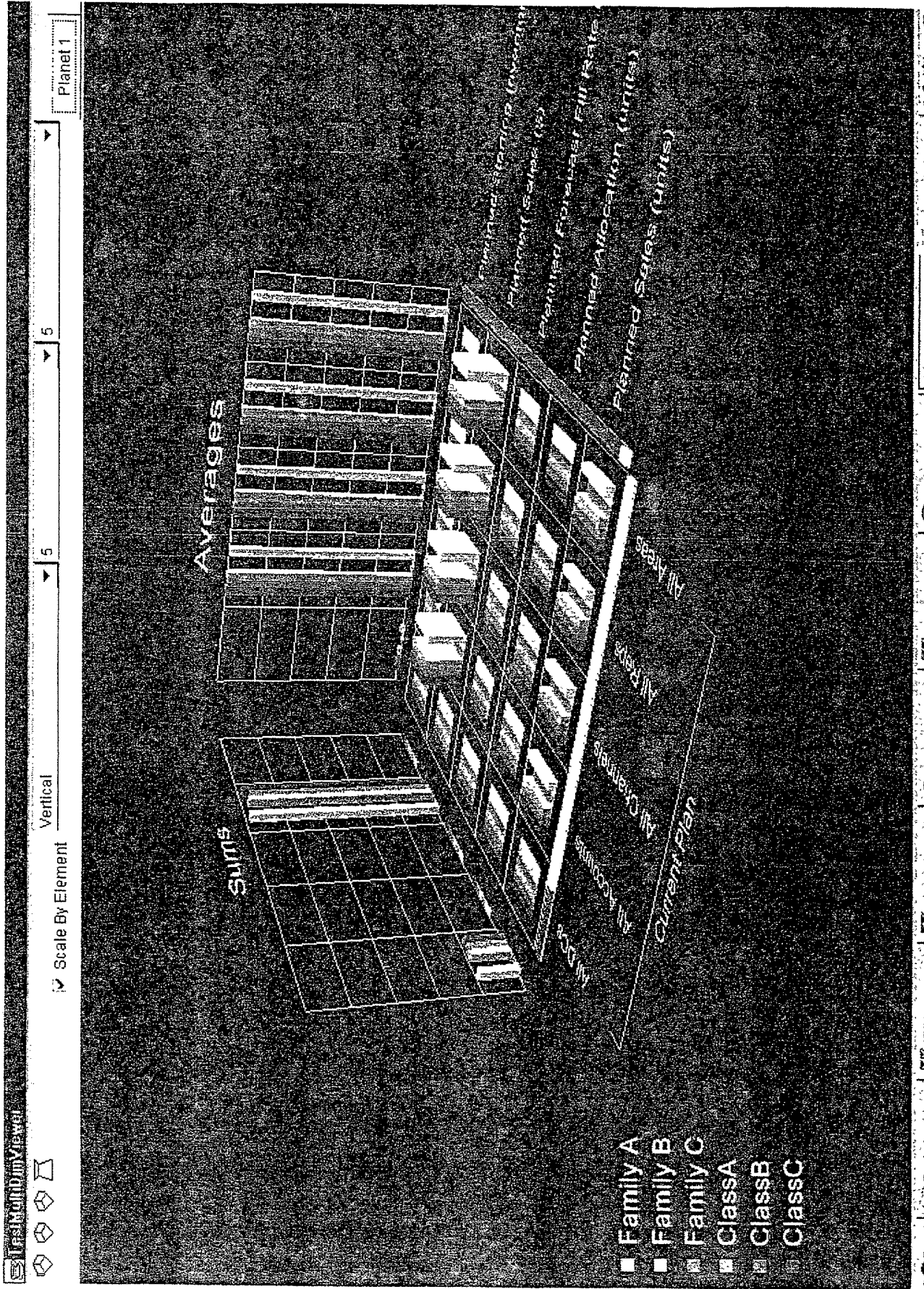
[illegible]

Figure 22

The Aggregation of Two Cell Hierarchy Values

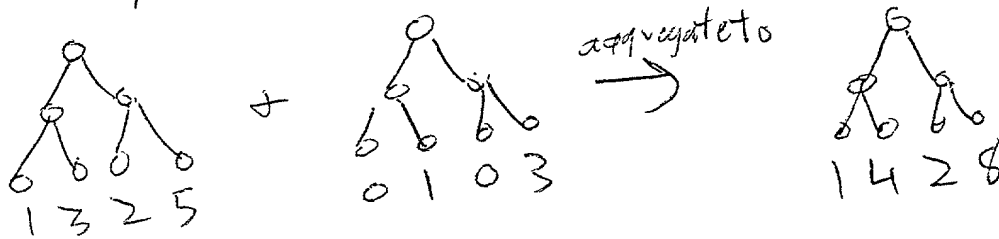


Figure 23

Cell Hierarchy

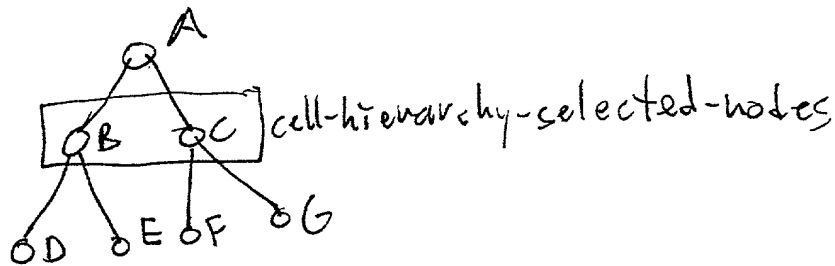


FIG 24

